Desarrollo ágil de software

Una introducción a las metodologías ágiles de desarrollo de software

Un artículo introductorio para comentar las características de las metodologías de desarrollo ágil de software.

Contenido

El desarrollo de programas es una disciplina que todos relacionamos en forma directa con el progreso, las mejoras en la productividad, y mucha gente inteligente trabajando duro y generando importantes beneficios para las empresas y toda la sociedad. Pero al mismo tiempo observamos que muchas veces los proyectos de desarrollo de software sufren retrasos y no se obtienen los resultados esperados pese al talento y esfuerzo puestos en acción por parte de analistas, programadores y usuarios para que “el nuevo sistema” funcione correctamente en tiempo y forma.

> ***¿Por qué tantos proyectos de desarrollo de software no se terminan a tiempo, cuestan más que lo presupuestado originalmente, tienen problemas de calidad serios y generan menor valor que el esperado?***

Este interrogante fue uno de los que se formularon Ken Schwaber, Jeff Sutherland y otros profesionales expertos en el desarrollo de software cuando se reunieron en febrero de 2001 para analizar el problema y decidieron redactar un “Manifiesto Ágil”. Se trató de un compromiso público en buscar nuevas y mejores formas de desarrollar software poniendo énfasis en las personas y sus interacciones, la colaboración y la respuesta continua al cambio, explorando nuevas formas de hacer las cosas, y compartiendo experiencias -- dando origen a una nueva comunidad de profesionales que explora sistemáticamente nuevas alternativas frente al modo tradicional de desarrollar software.

Desarrollo “tradicional” de software

La forma tradicional de desarrollar software se basa en procesos predefinidos con documentación muy precisa, y una detallada planificación inicial que debe seguirse estrictamente. Esta forma de trabajar surgió naturalmente hace unos cincuenta años como una adaptación del manejo de proyectos de ingeniería, que era lo más parecido a desarrollar programas que se conocía en ese momento, y funcionó razonablemente bien en un comienzo. También es necesario tener en cuenta que los ordenadores era enormemente caros, la mayor parte de la inversión informática se la llevaban los equipos y por esta razón los programas se hacían a medida para unas máquinas que se adquirian, no lo olvidemos, para realizar unas tareas muy concretas.

Pero los proyectos de desarrollo de software en la actualidad incluyen desafíos muy diferentes a los que se presentan al construir puentes y casas, por lo que no sorprende que los métodos tradicionales de desarrollo de software estén en crisis.

Tradicionalmente los proyectos se dividen en etapas bien diferenciadas: *Análisis de Factibilidad, Análisis de Requerimientos, Diseño, Programación, y Testeo.*

Generalmente se trata de que haya retroalimentación (*feedback* en inglés) entre las etapas contiguas, de tal forma de que, por ejemplo, haya un momento en que se mejoren los Requerimientos en base a comentarios, sugerencias y necesidades de los responsables del Diseño. Sin embargo, esta forma de desarrollar software genera muy serios problemas, debido a que al comienzo del proyecto, que es cuando menos se conocen las características del problema que resolver, se toman las decisiones de mayor relevancia e impacto en el resto del proyecto.

Como las chances de tomar decisiones erróneas al comienzo del proyecto son generalmente mayores que cuando ya se ha trabajado un tiempo en el proyecto, muchas veces proyectos no cumplen sus objetivos, no se terminan a tiempo, o resultan mucho más caros que lo presupuestado. En particular, esto ocurre frecuentemente en los casos en que el grupo de desarrollo necesita crear algo totalmente nuevo o de características específicas que nadie ha creado aún . . . lo cual es cierto en la mayoría de los proyectos de software, porque en caso contrario, la organización compraría directamente un producto o sistema ya desarrollado por otra empresa.

Como propuesta de solución a estos problemas han surgido una serie de "métodos ágiles" de desarrollo de software y manejo de proyectos en general, y cuyas principales características mencionaremos a continuación.

Características del Desarrollo Ágil

En los proyectos con Desarrollo Agil se busca que todos los recursos se empleen en la creación del mejor software que satisfaga las necesidades del cliente. Esto significa que todos los que forman parte del equipo de trabajo se concentran únicamente en tareas y procesos que agregan valor al cliente del producto que se está creando, mejorando o implementando. Adicionalmente, los usuarios o clientes reciben periódicamente prototipos o versiones en funcionamiento del producto a medida que se va construyendo, lo cual les permite evaluar el trabajo realizado, advertir sobre problemas que se detecten, y sugerir mejoras o funcionalidad valiosa que no se había considerado originalmente (ya sea por olvido, o porque la nueva funcionalidad se inspira en la experiencia de evaluar el producto mientras se está construyendo)

> **La distinción entre las tareas relevantes y los que no agregan valor se consigue a través de la creación de contextos con alto nivel de empowerment y feedback.**

El *empowerment* consiste en otorgar autonomía para tomar decisiones al equipo de desarrollo, y genera un clima de sinergia grupal que permite al grupo avanzar a pesar de las complicaciones y dificultades que ocurren habitualmente en los proyectos –de allí que uno de los métodos de trabajo más populares se haya bautizado con el nombre "scrum", ya que la imagen de los jugadores de rugby empleando su energía en avanzar todos juntos es muy aplicable a los equipos de trabajo que utilizan esta metodología.

El *feedback* constante y presente en varios niveles permite el desarrollo incremental y el crecimiento adaptativo de la programación, así también como una mejora constante en la forma de

trabajo de los equipos, lo que permite detectar problemas y resolverlos antes de que desaten crisis que afecten la calidad o el tiempo y costo del desarrollo. Los principales tipos de feedback ocurren a nivel producto, procesos y código

Periódicamente el cliente evalúa el estado real del software que se está creando, lo que asegura que lo entregado al final del proyecto coincidirá con lo esperado. Esto se consigue a través de un desarrollo incremental: el producto puede probarse desde las primeras semanas y meses del proyecto al menos en cuanto a su funcionalidad más básica, que luego va creciendo y mejorando –es por esto que se dice que desde el comienzo el producto ya tiene dentro su ADN, del mismo modo que ocurre con la gestación de los seres vivos en la Naturaleza.

A nivel procesos se realizan frecuentes reuniones retrospectivas donde los integrantes de los equipos comentan y discuten en profundidad tanto sus aciertos (para poder repetirlos y convertilos en hábitos), así también como el trabajo que no se realizó correctamente o no llevó al equipo a obtener los resultados esperados.

Adicionalmente los programadores suelen trabajar mucho en equipo y también por parejas, revisando juntos el código y resolviendo problemas en lugar de tratar de cubrirlos, lo que repercute en un producto de mejor calidad, mejor documentado, y simple de mantener.

Ricardo Colusso
http://crealogar.blogspot.com

Comentarios

Accede para escribir un comentario.

Igancio Sánchez-Ferrero

¿Exceso de firmas?

Veo que tu knol está firmado tres veces. Una es la automática en el lateral, la segunda al principio del texto y la tercera al final ¿Quizá se podría suprimir una?

Última modificación: 05/09/2009 13:54
Informe de comentarios abusivos
+2 Ver/publicar las respuestas (2) a este comentario ▼